

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-301911

(43)Date of publication of application : 13.11.1998

(51)Int.Cl.

G06F 15/163
G06F 13/00

(21)Application number : 09-112781

(71)Applicant : FUJITSU LTD

(22)Date of filing : 30.04.1997

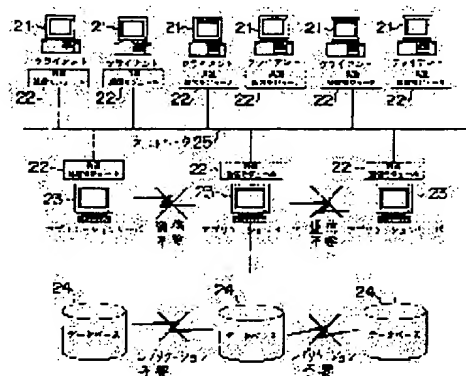
(72)Inventor : HARA HIROTAKE
MATSUZUKA TAKAHIDE
KANETANI NOBUYUKI
OKUBO TAKAO
UEHARA SANPACHI

(54) NETWORK COMPUTER SYSTEM LED BY CLIENT AND ITS METHOD

(57)Abstract:

PROBLEM TO BE SOLVED: To easily execute the updating and addition of information in a distributed client-server system.

SOLUTION: Each client 21 can be connected to an optional server 23 by using a common communication module 22 and is allowed to utilize the functions of respective servers 23 and access the information of respective data bases 24 by switching its connection. Thereby it is basically unnecessary to execute communication between servers 23 and replication between data bases 24.



LEGAL STATUS

[Date of request for examination] 21.09.2000

[Date of sending the examiner's decision of rejection] 30.11.2004

[Kind of final disposal of application other than the
examiner's decision of rejection or application
converted registration]

[Date of final disposal for application]

[Patent number] 3652834

[Date of registration] 04.03.2005

[Number of appeal against examiner's decision of
rejection] 2005-00012[Date of requesting appeal against examiner's decision
of rejection] 04.01.2005

[Date of extinction of right]

BEST AVAILABLE COPY

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平10-301911

(43)公開日 平成10年(1998)11月13日

(51)Int.Cl.⁴

識別記号

F I

G 0 6 F 15/163

G 0 6 F 15/16

3 1 0 R

13/00

3 5 7

13/00

3 5 7 Z

審査請求 未請求 請求項の数23 O L (全 26 頁)

(21)出願番号 特願平9-112781

(22)出願日 平成9年(1997)4月30日

(71)出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番
1号

(72)発明者 原 裕貴

神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

(72)発明者 松塚 貴英

神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

(74)代理人 弁理士 大管 義之 (外1名)

最終頁に続く

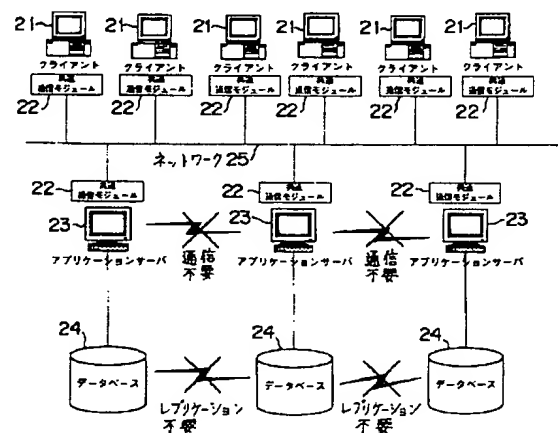
(54)【発明の名称】 クライアント主導のネットワーク・コンピューティングシステムおよび方法

(57)【要約】

【課題】 分散クライアント-サーバシステムにおいて、情報の更新や追加を容易に行うことが課題である。

【解決手段】 クライアント21は、サーバ23に共通の通信モジュール22を用いて、任意のサーバ23に接続することができ、接続を切り替えることによって、各サーバ23の機能を利用したり、各データベース24の情報にアクセスしたりする。したがって、サーバ23間での通信やデータベース24間でのレプリケーションは、基本的に行う必要がない。

ネットワーク・コンピューティングシステムの構成図



【特許請求の範囲】

【請求項1】 通信ネットワークを介した分散システムにおいて情報を管理するネットワーク・コンピューティングシステムであって、

前記ネットワーク上のクライアントと複数のサーバとの通信を制御し、該複数のサーバに共通にアクセス可能な共通通信手段と、

前記クライアントが前記複数のサーバのうちの第1のサーバに接続したとき、接続状態を保持したまま、必要に応じて前記複数のサーバのうちの第2のサーバに接続し、該クライアントに該第2のサーバからのサービスを受けさせる接続管理手段とを備えることを特徴とするネットワーク・コンピューティングシステム。

【請求項2】 前記クライアント側に設けられ、前記第1のサーバ内の情報と前記第2のサーバ内の情報との間のリンク情報を、該第1のサーバから受け取って管理するリンク情報管理手段をさらに備え、前記接続管理手段は、前記リンク情報管理手段に保持された前記リンク情報を用いて、ネットワーク透過的に前記第2のサーバに接続することを特徴とする請求項1記載のネットワーク・コンピューティングシステム。

【請求項3】 前記複数のサーバが、前記クライアントの前記リンク情報管理手段を利用して情報の交換を行い、前記ネットワーク上での複雑な処理を実現することを特徴とする請求項2記載のネットワーク・コンピューティングシステム。

【請求項4】 前記接続状態は、前記第1のサーバ内のオブジェクトと第2のサーバ内のオブジェクトとの間でメソッド実行要求を送信することが可能な状態を表し、該第1のサーバからのメソッド実行要求を前記クライアントを介して該第2のサーバに送ることで、該第1のサーバと第2のサーバの間で通信することなく、前記クライアントが該第2のサーバからのサービスを受けることを特徴とする請求項1記載のネットワーク・コンピューティングシステム。

【請求項5】 通信ネットワークを介した分散システムにおいて情報を管理するネットワーク・コンピューティングシステムであって、

前記ネットワーク上の第1のサーバとクライアントとの通信を制御し、該クライアントからのアクセスを受け付ける共通通信手段と、

前記クライアントが前記第1のサーバとの接続状態を保持したまま、必要に応じて第2のサーバに接続し、該第2のサーバからのサービスを受けるためのリンク情報を、該クライアントに送信する接続管理手段とを備えることを特徴とするネットワーク・コンピューティングシステム。

【請求項6】 前記ネットワーク上に分散した任意のサーバ上に、オブジェクトを生成するオブジェクト生成手段と、オブジェクト間のリンクを生成するリンク設定手

段とをさらに備え、接続管理手段は、前記第1のサーバ内のオブジェクトと前記第2のサーバ内のオブジェクトとの間のリンクに関する情報を、前記リンク情報として、前記クライアントに送信することを特徴とする請求項5記載のネットワーク・コンピューティングシステム。

【請求項7】 前記リンク設定手段は、双方向のリンクを生成することを特徴とする請求項6記載のネットワーク・コンピューティングシステム。

10 【請求項8】 各オブジェクトのクラス間の関係を定義するクラス定義情報を保持し、該クラス定義情報に基づいて前記オブジェクト間のリンクを管理するクラス情報管理手段をさらに備えることを特徴とする請求項6記載のネットワーク・コンピューティングシステム。

【請求項9】 前記クラス情報管理手段は、クラス間の階層構造を持つような前記クラス定義情報を保持することを特徴とする請求項8記載のネットワーク・コンピューティングシステム。

20 【請求項10】 前記クラス情報管理手段は、企業活動に関する組織、個人、プロセス、および文書のうち少なくとも1つ以上のクラスを定義し、前記分散システム上で企業モデルを構築することを特徴とする請求項9記載のネットワーク・コンピューティングシステム。

【請求項11】 前記ネットワーク上における各オブジェクトの所在場所に関するディレクトリ情報を保持し、該ディレクトリ情報を管理するディレクトリ情報管理手段をさらに備えることを特徴とする請求項6記載のネットワーク・コンピューティングシステム。

30 【請求項12】 前記ディレクトリ情報管理手段は、一部のオブジェクトのディレクトリ情報のみを保持することで、該ディレクトリ情報の量を削減することを特徴とする請求項11記載のネットワーク・コンピューティングシステム。

【請求項13】 前記ディレクトリ情報の一部をサーバ間でレプリケートするレプリケーション手段をさらに備えることを特徴とする請求項11記載のネットワーク・コンピューティングシステム。

40 【請求項14】 各サーバ内のオブジェクトの一部をサーバ間でレプリケートするレプリケーション手段をさらに備えることを特徴とする請求項6記載のネットワーク・コンピューティングシステム。

【請求項15】 レプリケートされたオブジェクトの更新処理時に、該オブジェクトのマスターオブジェクトに自動的に接続を切り替えて、該マスターオブジェクトを更新する手段をさらに備えることを特徴とする請求項14記載のネットワーク・コンピューティングシステム。

50 【請求項16】 前記ネットワーク上における各サーバの所在場所に関する情報をサーバ間で共有させ、該所在場所に関する情報を用いて各オブジェクトの管理を行う手段をさらに備えることを特徴とする請求項6記載のネ

ットワーク・コンピューティングシステム。

【請求項17】 前記分散システムに、前記オブジェクト生成手段とリンク設定手段を持つ新しいサーバを登録する新サーバ設定手段をさらに備えることを特徴とする請求項6記載のネットワーク・コンピューティングシステム。

【請求項18】 ユーザ情報に関するオブジェクトを利用して、前記分散システムにおけるユーザ管理を行う手段をさらに備えることを特徴とする請求項6記載のネットワーク・コンピューティングシステム。

【請求項19】 前記リンク情報を用いて情報検索サービスを行う手段をさらに備えることを特徴とする請求項6記載のネットワーク・コンピューティングシステム。

【請求項20】 通信ネットワークを介した分散システムにおいて情報を管理するためのプログラムを記録した記録媒体であって、

前記ネットワーク上のクライアントと複数のサーバとの通信を制御し、該複数のサーバに共通にアクセスする機能と、

前記クライアントが前記複数のサーバのうちの第1のサーバに接続したとき、接続状態を保持したまま、必要に応じて前記複数のサーバのうちの第2のサーバに接続し、該クライアントに該第2のサーバからのサービスを受けさせる機能とをコンピュータに実現させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項21】 通信ネットワークを介した分散システムにおいて情報を管理するためのプログラムを記録した記録媒体であって、

前記ネットワーク上の第1のサーバとクライアントとの通信を制御し、該クライアントからのアクセスを受け付ける機能と、

前記クライアントが前記第1のサーバとの接続状態を保持したまま、必要に応じて第2のサーバに接続し、該第2のサーバからのサービスを受けるためのリンク情報を、該クライアントに送信する機能とをコンピュータに実現させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項22】 通信ネットワークを介した分散システムにおいて情報を管理するネットワーク・コンピューティング方法であって、

前記ネットワーク上のクライアントが、該ネットワーク上の複数のサーバに共通にアクセスするように通信を制御し、

前記クライアントが前記複数のサーバのうちの第1のサーバに接続したとき、接続状態を保持したまま、必要に応じて前記複数のサーバのうちの第2のサーバに接続するように通信を制御して、該クライアントに該第2のサーバからのサービスを受けさせることを特徴とするネットワーク・コンピューティング方法。

【請求項23】 前記第2のサーバからのサービスを受けるためのリンク情報を、前記第1のサーバから前記クライアントに送信することを特徴とする請求項22記載のネットワーク・コンピューティング方法。

【発明の詳細な説明】

【0001】

【発明が属する技術分野】本発明は、ネットワークを介した分散クライアント-サーバシステムにおいて、情報の更新や追加を行うネットワーク・コンピューティングシステムおよびその方法に関する。

【0002】

【従来の技術】従来の情報システムにおいては、単一の巨大なデータベースに対してトランザクションベースで処理を行うのが主流であった。しかし、今日では、計算機システムやネットワークの発達によって、広範囲に分散したシステム間で頻繁に情報の更新、追加が行えるような情報システムを構築することが求められている。また、LAN (local area network) とグローバルネットワークの融合により、ネットワークを中心とするネットワーク・コンピューティングという形の情報処理が実現されつつある。

【0003】図19は、従来のクライアント-サーバシステムの構成図である。図19のシステムは、クライアント1、アプリケーションサーバ5 (アプリケーションサーバA、B、C)、およびデータベース6から成る。アプリケーションサーバA、B、Cは、データベース6を共有し、クライアント1と通信するために、それぞれ、サーバA専用通信モジュール2、サーバB専用通信モジュール3、サーバC専用通信モジュール4を搭載している。

【0004】クライアント1は、サーバA専用通信モジュール2、サーバB専用通信モジュール3、サーバC専用通信モジュール4のうちのいずれかを搭載し、対応する通信モジュールを有する特定のアプリケーションサーバ5に接続して、データベース6の更新等を行う。

【0005】さらに、近年のデータベース6の巨大化やシステムの分散化に伴って、図20に示すように、データベース6が分散して配置されるようになってきた。図20のシステムでは、アプリケーションサーバA、B、Cは、それぞれ、個別のデータベース6を管理している。このシステムにおいても、クライアント1は、サーバA専用通信モジュール2、サーバB専用通信モジュール3、およびサーバC専用通信モジュール4のいずれかをを用いて、特定のアプリケーションサーバ5にのみ接続する。

【0006】このため、クライアント1が接続先のアプリケーションサーバ5とは異なる他のサーバ5のデータベース6の情報にアクセスする際には、必要に応じてデータベース6間でレプリケーションが行われたり、アプリケーションサーバ5間でRPC (remote procedure c

all)が行われたりしている。レプリケーションの場合は、データベース6間でデータのコピーが行われ、RPCの場合は、アプリケーションサーバ5間で関数呼び出しが行われて、必要なデータがやり取りされる。

【0007】図19および図20のシステムに共通しているのは、クライアント1が通信モジュール2、3、4を用いて、特定のアプリケーションサーバ5と接続するように設計されていることである。

【0008】また、近年発展しているインターネット上においては、WWW(world wide web)が開発され、情報発信の手段として広く利用されている。WWWの特徴は、http(hypertext transfer protocol)という共通のプロトコルによって任意のクライアント(WWWブラウザ)から任意のサーバ(WWWサーバ)に接続できることである。また、WWWは、サーバからクライアントへ送られるファイルであるHTML(hypertext markup language)文書中に、他のサーバ上のHTML文書へのリンクを埋め込むハイパーリンク機能を持っている。

【0009】図21は、WWWにおけるクライアントおよびサーバの動作フローチャートである。クライアントが動作を開始し、ユーザがその画面上のリンクをクリックすると(ステップS1)、クライアントがhttpを用いて該当するサーバに接続する(ステップS2)。

【0010】これにより、サーバから該当するHTML文書がクライアントに送信され(ステップS3)、クライアントは、サーバとの接続を切断して(ステップS4)、HTML文書の情報を画面上に表示する(ステップS5)。ユーザが表示された画面上のリンクを再びクリックすると、ステップS1からS5までの動作が繰り返される。

【0011】

【発明が解決しようとする課題】しかしながら、上述した従来の情報システムには次のような問題がある。図19または図20のような従来のクライアント-サーバシステムでは、クライアント1は特定のサーバ5としか接続できないため、ネットワーク上に分散した他のサーバ5の機能を直接利用することはできない。

【0012】このため、クライアント1の接続先のサーバ5は、同時に、他のサーバ5と接続して必要な処理を依頼しなければならない、サーバ5間でトランザクションが多発する。また、データベース6間のレプリケーションを必要とする場合も、同様に、処理が煩雑になる。

【0013】このように、サーバ5間やデータベース6間における処理の制御が複雑になるため、WWWのように、サーバ5やデータベース6が全世界的に広がったような巨大な分散システムを構築することは不可能である。

【0014】また、従来のWWWでは、クライアントからの要求に対してHTML文書を送るという単純な機能

しか提供されていないという問題がある。特に、httpでは、1つのページの送信毎にセッションが切れてしまうため、通信状態(接続状態)を保持することができない。したがって、WWWの基本機能のみでは、本格的なインタラクションを行うクライアント-サーバシステムを開発することは困難である。

【0015】さらに、WWWにおいては、ページやリンクの作成/変更は個々のユーザに任されているため、WWW全体がどのような構造になっているかは把握不能であり、例えば、以下のような問題がある。

【0016】1. 新しいページの実体データ(オブジェクト)を、分散した任意のサーバ上に作成することは難しい。

2. 新しいオブジェクトの登録の際に、既存のオブジェクトからのリンクが更新/管理されない。

【0017】3. オブジェクトの削除に伴いリンク先のオブジェクトがなくなってしまうこともある(デッドリンク)。

4. ユーザが欲しい情報を検索する手段として、WWWロボットのようなプログラムが開発されているが、オブジェクト間に構造がないために、すべてのオブジェクトに対してキーワードによる探索を行わなければならない。

【0018】本発明の課題は、ネットワークを介した分散クライアント-サーバシステムにおいて、情報の更新や追加を容易に行うことができるネットワーク・コンピューティングシステムおよびその方法を提供することである。

【0019】

【課題を解決するための手段】図1は、本発明のネットワーク・コンピューティングシステムの原理図である。図1のネットワーク・コンピューティングシステムは、通信ネットワークを介した分散システムにおいて、分散した情報を管理する。

【0020】本発明の第1の原理において、クライアント側は、共通通信手段11と接続管理手段12を備える。共通通信手段11は、前記ネットワーク上のクライアントと複数のサーバとの通信を制御し、それらの複数のサーバに共通にアクセスすることができる。接続管理手段12は、上記クライアントが上記複数のサーバのうちの第1のサーバに接続したとき、接続状態を保持したまま、必要に応じて上記複数のサーバのうちの第2のサーバに接続し、上記クライアントに第2のサーバからのサービスを受けさせる。

【0021】共通通信手段11は、従来のように特定のサーバとの通信を行うものではなく、ネットワーク上のすべてのサーバと共通に通信することができる。例えば、共通通信手段11の制御により、クライアントが第1のサーバに接続しているときに、第2のサーバに処理を依頼する必要が生じたとする。このとき、接続管理手

段12の制御により、クライアントは第1のサーバとの接続状態を保持したまま、第2のサーバに接続して、第2のサーバからのサービスを受ける。

【0022】第1の原理によれば、クライアントは、特定のサーバを経由することなく、直接適切なサーバに接続して、情報提供等のサービスを受けることができる。また、情報の更新や追加に伴うサーバ間の処理の依頼も、クライアントを介して自動的に行うことができる。

【0023】また、本発明の第2の原理において、サーバ側は、共通通信手段13と接続管理手段14を備える。共通通信手段13は、上記ネットワーク上の第1のサーバとクライアントとの通信を制御し、そのクライアントからのアクセスを受け付ける。接続管理手段14は、上記クライアントが第1のサーバとの接続状態を保持したまま、必要に応じて第2のサーバに接続し、第2のサーバからのサービスを受けるためのリンク情報を、上記クライアントに送信する。

【0024】共通通信手段13は、従来のようにサーバ毎に専用のものではなく、ネットワーク上のすべてのサーバに共通している。例えば、共通通信手段13の制御により、クライアントが第1のサーバに接続しているときに、第2のサーバに処理を依頼する必要が生じたとする。このとき、接続管理手段14の制御により、第2のサーバにアクセスするためのリンク情報がクライアントに送られる。クライアントは、第1のサーバとの接続状態を保持したまま、受け取ったリンク情報を用いて第2のサーバに接続し、第2のサーバからのサービスを受ける。

【0025】第2の原理によれば、クライアントは、接続中のサーバから次のサーバへのリンク情報を受け取ることで、ネットワーク上の任意のサーバにアクセスして、情報提供等のサービスを受けることができる。また、第1の原理と同様に、情報の更新や追加に伴うサーバ間の処理の依頼も、クライアントを介して自動的に行うことができる。

【0026】このように、本発明のシステムによれば、基本的にサーバ間で通信する必要がなくなり、サーバ側の通信制御が簡単になる。したがって、大規模な分散システムを比較的容易に構築することが可能となる。

【0027】例えば、図1の共通通信手段11、共通通信手段13は、それぞれ、次に説明する図2におけるクライアント21の共通通信モジュール22、サーバ23の共通通信モジュール22に対応する。また、接続管理手段12、接続管理手段14は、それぞれ、後述する図4におけるクライアント21の接続管理モジュール41、サーバ23の接続管理モジュール41に対応する。

【0028】

【発明の実施の形態】以下、図面を参照しながら、本発明の実施の形態を詳細に説明する。図2は、本実施形態のネットワーク・コンピューティングシステムの基本ア

ーキテクチャを示す構成図である。図2において、各クライアント21と各アプリケーションサーバ23は、LAN等の任意の通信ネットワーク25により結合され、共通の通信モジュール22を持っている。また、各サーバ23は、それぞれ、個別のデータベース24を管理している。

【0029】このシステムでは、共通通信モジュール22を用いて、クライアント21から任意のサーバ23に接続することができる。これらのサーバ23は、例えば、1つの企業内で部門毎に配置される均質な（ホモジニアスな）ものでもよく、それぞれ機能が異なる非均質な（ヘテロジニアスな）ものでもよい。

【0030】クライアント21は、接続を切り替えることによって、各サーバ23の機能を利用したり、各データベース24の情報にアクセスしたりするので、原則として、サーバ23間での通信やデータベース24間でのレプリケーションは行う必要がない。共通通信モジュール22は、例えば、ジャバ（J A V A）アプレットなどを用いて、クライアント21が最初にサーバ23に接続したときに、クライアント21にダウンロードすることもできる。

【0031】ジャバとは、インタプリタベースのネットワーク対応プログラム言語であり、ジャバアプレットとは、ジャバでコーディングされたプログラムのことである。ジャバアプレットは、どのオペレーティング・システム（OS）でも実行することができ、例えば、サーバ23からダウンロードされて、クライアント21上で実行される。

【0032】図3は、クライアント21またはサーバ23として用いられる情報処理装置の構成図である。図3の情報処理装置は、CPU（中央処理装置）31、メモリ32、入力装置33、出力装置34、外部記憶装置35、媒体駆動装置36、ネットワーク接続装置37を備え、それらの各装置はバス38により互いに結合されている。

【0033】CPU31は、メモリ32に格納されたプログラムを実行して、クライアント21またはサーバ23の各処理を実現する。メモリ32としては、例えばROM（read only memory）、RAM（random access memory）等が用いられる。

【0034】入力装置33は、例えばキーボードやポインティングデバイス等に相当し、ユーザからの指示の入力に用いられる。出力装置34は、例えば表示装置やプリンタ等に相当し、ユーザへ提示する画面（ビュー）の表示やその他の情報出力に用いられる。

【0035】外部記憶装置35は、例えば、磁気ディスク装置、光ディスク装置、光磁気ディスク装置等であり、プログラムやデータを保存することができる。また、データベース24としても使用することができる。

【0036】媒体駆動装置36は、可搬記録媒体39を

駆動し、その記憶内容にアクセスする。可搬記録媒体39としては、メモリカード、フロッピーディスク、CD-ROM (compact disk read only memory)、光ディスク、光磁気ディスク (magneto-optical disk) 等、任意のコンピュータ読み取り可能な記録媒体を使用することができる。この可搬記録媒体39には、データのほかに、クライアント21またはサーバ23の各処理を行うプログラムが格納される。

【0037】ネットワーク接続装置37は、ネットワーク25に接続され、通信に伴うデータ変換等を行う。クライアント21およびサーバ23は、ネットワーク接続装置37を介して、ネットワーク25から必要な情報を受け取ることができる。

【0038】次に、図4から図6までを参照しながら、クライアント21が複数のサーバ23を意識することなしに、ネットワーク透過的に必要なサーバ23に接続する方法を説明する。ここでは、各データベース24のデータをオブジェクトとして管理し、クライアント21からサーバ23へのネットワーク透過的な接続を実現するために、オブジェクト間のリンク情報を利用する。

【0039】オブジェクトとは、オブジェクト指向データベースにおける情報の単位であり、データとそれに適用される手続きであるメソッドとの組みを有する。オブジェクトに定義されたメソッドを実行することで、様々な情報処理が実現される。

【0040】図4は、クライアント21が1つのサーバ23に接続している状態を示している。図4において、クライアント21およびサーバA、Bの接続管理モジュール41は、図2の共通通信モジュール22に含まれる。また、リンク情報管理モジュール42を、共通通信モジュール22に含めておいてもよい。

【0041】今、クライアント21はサーバAに接続しており、サーバA上のオブジェクト44に関する情報が、ユーザビュー43として、クライアント21の画面に表示されている。

【0042】オブジェクト44の論理IDは、“A／／001”であり、これは、サーバAにおけるオブジェクトIDが“001”であることを表している。また、オブジェクト44が属するクラスは“組織”であり、そのアトリビュート（属性）として、組織名“ソフトウェア研究部”、部コード“1111”、および電話番号“2222”が定義されている。

【0043】また、オブジェクト44には、関連する文書である“研究計画”および“月報”のオブジェクトへのリンクとして、他のサーバBのオブジェクト45の論理ID“B／／001”、およびサーバCのオブジェクト（不図示）の論理ID“C／／002”が定義されている。

【0044】このように、各オブジェクトが保持するリンクは、同一のサーバ23上のオブジェクト間のみなら

ず、異なるサーバ23上のオブジェクト間にも張ることができるようにしている。このリンクの情報は、クライアント21がオブジェクトにアクセスしたときに、接続管理モジュール41を介して、サーバ23からクライアント21のリンク情報管理モジュール42に送られる。

【0045】また、メソッドとしては、アトリビュート変更とワークフロー実行が定義されている。ワークフローとは、組織内における作業をモデル化するためのソフトウェアであるワークフローツールに対する操作を表し、ここでは、ユーザによる“月報”の作成作業を指している。

【0046】オブジェクト44が持つこれらの情報は、ユーザビュー43として、クライアント21のユーザに提示される。また、リンク情報管理モジュール42は、研究計画文書へのリンク“B／／001”と月報文書へのリンク“C／／002”とを、それぞれ、ユーザビュー43上の詳細ボタン48、49と関連付けて保持している。

【0047】このユーザビュー43上で、ユーザが操作を行うことにより、サーバAのオブジェクト44のメソッドが実行されたり、他のサーバB、Cのオブジェクトが自動的にアクセスされたりする。

【0048】例えば、表示された部コードを修正して、修正ボタン46を押す（クリックする）ことによって、その情報がサーバAに送られ、オブジェクト44のアトリビュート変更というメソッドが実行される。また、ワークフローの実行ボタン47を押すことによって、ワークフローツールが起動される。ワークフローツールは、サーバA上にあってもよく、他のサーバ23上にあってもよい。

【0049】また、研究計画文書の詳細ボタン48を押すことによって、図5に示すように、それに関連付けられたオブジェクトへのアクセスが、リンク情報管理モジュール42に依頼される。この例では、リンク先はサーバBのオブジェクト45なので、クライアント21の接続管理モジュール41からサーバBの接続管理モジュール41に対して接続依頼が行われる。

【0050】サーバBの接続管理モジュール41は、論理ID“B／／001”からオブジェクト45へのポインタを求め、クライアント21との接続を確立する。接続が確立された後、オブジェクト45に関する表示データ／リンク情報が、クライアント21に送信される。

【0051】これにより、ユーザビュー43は、図6に示すように変更される。図6のユーザビュー50においては、オブジェクト45のクラスを表す“文書”、名称を表す“研究計画”、作成日を表す“1996年12月10日”、および作成者を表す“田中一郎”が表示されている。

【0052】また、リンク情報管理モジュール42は、

オブジェクト45が所属する組織を表すオブジェクト44の論理IDを、所属組織の詳細ボタン51に関連付けられたリンクとして保持する。

【0053】ユーザビュー50において、修正ボタン51が押されると、オブジェクト45のアトリビュート変更というメソッドが実行され、参照ボタン52が押されると、文書参照というメソッドが実行され、修正ボタン53が押されると、文書修正というメソッドが実行される。

【0054】また、詳細ボタン54が押されると、サーバAのオブジェクト44の情報が再び表示される。このように、オブジェクト44のリンク“B／／001”とオブジェクト45のリンク“A／／001”は、お互いをリンク先とする相互リンク情報を構成している。

【0055】このように、本実施形態のシステムでは、クライアント21とサーバAとのセッションと、クライアント21とサーバBとのセッションが、1つの接続状態（コンテキスト）として管理される。ここで、接続状態とは、サーバA内のオブジェクトとサーバB内のオブジェクトとの間で、メソッドの実行要求や実行結果をやり取りすることが可能な通信状態を表す。

【0056】サーバAからのメソッド実行要求を、クライアント21を介してサーバBに送ることで、サーバAとサーバBの間で通信することなく、クライアント21がサーバBからのサービスを受けることができる。このようなことが可能であるのは、クライアント21側とサーバ23側のそれぞれの通信モジュール22が、標準化／共通化されているためである。

【0057】また、クライアント21を介して多数のサーバ23の間で情報を交換することにより、サーバ23間での通信を行うことなく、ネットワーク25上で複雑な処理を実現することができる。

【0058】次に、サーバ23が切り替わったときの接続状態の保持や、サーバ23間の情報交換をクライアント21を通じて行う方法について説明する。例えば、あるサーバ23から別のサーバ23に切り替わるときに、ユーザIDおよびパスワードや現在の状態等の付加的な情報を、新しいサーバ23に送る例を考える。ここで、現在の状態とは、現在、クライアント21上でどのようなビューが開かれているか等の処理状況に関する情報を指す。

【0059】まず、図7に示すように、従来のクライアント-サーバシステムと同様に、サーバ23同士が接続して情報を交換する方法が考えられる。しかし、このような方法では、多数のクライアント21からのアクセスを同時に処理する場合、サーバ23間でトランザクションが多発し、サーバ23の処理が煩雑になる。このため、システム全体の性能にも悪影響を及ぼす。

【0060】そこで、本実施形態では、図8に示すように、クライアント21が新しいサーバ23に新規接続要

求を出すときに、サーバ23間の情報の受け渡しも行うようにする。この方法により、サーバ23間の複雑な通信処理がなくなり、巨大な分散システムの構築が可能となる。

【0061】図9は、ネットワーク・コンピューティングシステムにおけるモジュール構成の例を示している。ただし、図9のシステムにおいては、上述の共通通信モジュール22および接続管理モジュール41は省略されており、ネットワーク25上には、他のサーバ23を管理する管理サーバ61が配置されている。

【0062】管理サーバ61は、ディレクトリ情報66およびクラス定義情報67を有するデータベース62に接続され、ディレクトリ情報管理モジュール63、クラス情報管理モジュール64、および新サーバ設定モジュール65を有する。ディレクトリ情報管理モジュール63は、ディレクトリ情報を管理し、クラス情報管理モジュール64は、クラス情報を管理し、新サーバ設定モジュール65は、新しいサーバの設定を行う。

【0063】また、ネットワーク25上に分散した各サーバ23は、リンク設定モジュール70、オブジェクト生成モジュール71、レプリケーションモジュール72、および各種サービスモジュール73を含み、データベース24は、オブジェクト情報68およびレプリケーション情報69を有する。

【0064】オブジェクト生成モジュール71は、各オブジェクトを生成し、データベース24にオブジェクト情報68として格納する。リンク設定モジュール70は、生成されたオブジェクト情報68に、ネットワーク25上に分散したオブジェクト間のリンクを設定する。レプリケーション情報69、レプリケーションモジュール72、および各種サービスモジュール73については後述することにする。

【0065】クライアント21は、上述のリンク情報管理モジュール42と、画面上にビューを表示するビューモジュール74を有し、サーバ23にアクセスして、そのオブジェクトを参照したり、作成／更新したりする。

【0066】管理サーバ61、サーバ23、およびクライアント21の各モジュール自身もオブジェクトとして管理され、データおよびメソッドを持っている。したがって、各モジュールの機能は、対応するメソッドを実行することにより実現される。

【0067】サーバ23上で生成されたオブジェクトは、管理サーバ61上のディレクトリ情報管理モジュール63によって、ディレクトリ情報66を用いて管理される。ディレクトリ情報66は、ネットワーク25上における各オブジェクトの所在場所に関する情報を表し、例えば図10に示すように、各クラス毎に一意のオブジェクト名（またはオブジェクトID）と、オブジェクトアドレスのペアを保持する。

【0068】図10において、例えば、オブジェクト名

として“研究所”を持つオブジェクトアドレスは、研究所サーバ／／1001であることが記述されている。このアドレスは、オブジェクトの論理IDに相当する。また、オブジェクトアドレスとして、WWWにおけるURL (uniform resource locator) 等を用いてもよい。ここでは、各オブジェクト間のリンク情報は、ディレクトリ情報66としては管理されていない。

【0069】次に、サーバ23におけるリンク情報のメンテナンスについて説明する。従来は、分散したマシン上のオブジェクト間のリンクは構造化されていなかったが、クラス概念を導入することで、これらを構造化することができる。

【0070】クラスとは、それに属するオブジェクトの属性や、他のオブジェクトとの間のリンクを定義するもので、その情報はクラス定義情報67に保持される。クラス情報管理モジュール64は、クラス定義情報67の作成／更新等を行う。

【0071】例えば、図10のオブジェクト“研究所”のクラス名は“組織”である。また、図11のクラス定義情報67の例では、組織クラスと個人クラスとの間のリンクが定義されている。これらのリンクは、内部的には、一意な名称またはIDで管理される。各リンクに対しては任意の性質を定義することができ、例えば、必ず必要なリンクであることを示す情報や、複数のリンクがあってもよいことを示す情報等が、性質として設定される。

【0072】図11では、組織にはN人 (Nは1以上) の所属員がいて、逆に、ある個人は1つの組織に属するという関係が、リンクとして定義されている。このようなリンクを特に指定したくないオブジェクトは、リンクを持たない特別なルートクラスとして扱うものとする。リンクは片方向のみ定義してもよいが、本実施形態では、図11のような双方向のリンクまで含めて定義することができる。

【0073】このようなクラスの定義のための言語や図形は各種提案されており、クラスには、メソッドを定義することもできる。本発明では、クラスの情報のうちリンクに関する情報が特に重要な役割を果たし、属性やメソッドについては一般のオブジェクトと同様である。

【0074】クラスにおいて定義されたリンクを用いることによって、企業の組織のような階層構造を管理することもできる。図12は、企業活動のモデルをクラス定義を用いて表現した例を示している。

【0075】図12において、Location (場所) クラス81とServer (サーバ) クラス82の間、およびLocationクラス81とOrganization/Project (組織／プロジェクト) クラス83の間には、Located というリンクが設定され、Serverクラス82とOrganization/Projectクラス83の間、Organization/Projectクラス83とDocument (文書) クラス84の間、およびDocumentクラス84

とIndividual (個人) クラス85の間には、BelongTo というリンクが設定されている。

【0076】また、Organization/Projectクラス83とIndividualクラス85の間には、MemberOf というリンクが設定され、Organization/Projectクラス83とBusiness Function (業務) クラス86の間には、PurposeOf というリンクが設定されている。

【0077】Business Function クラス86は、いくつかのProcess (プロセス) クラス87を含み、Process クラス87とProcess クラス87の間には、Pre/Post というリンクが設定される。さらに、Process クラス87とDocumentクラス84の間には、Input/Output というリンクが設定され、Process クラス87とIndividualクラス85の間には、Charged というリンクが設定されている。

【0078】また、Locationクラス81、Organization/Projectクラス83には、その内部構造として階層を定義することができ、複数のProcess クラス87間に階層構造を定義することもできる。このような企業活動のモデルを分散システム上で構築し、管理することによって、業務の効率化や各種の業務アプリケーションの統一的な管理を行うことができる。

【0079】次に、具体例を参照しながら、図9のシステムの処理についてより詳細に説明する。以下の実施形態では、主として、WWWクライアント上で表示される組織オブジェクト情報の例が用いられている。

【0080】今、図13に示すように、ソフト研 (ソフトウェア研究部) サーバ91上の“ソフトウェア研究部”という組織のオブジェクト93を参照しているクライアント21が、所属員“山田”の情報を参照するため、そのリンクを指示したとする。このとき、そのリンク情報により、クライアント21が、“山田”のオブジェクトが存在するソフト開発 (ソフトウェア開発部) サーバ92に接続し、“山田”に関するオブジェクト94の情報を表示する。ただし、ソフト研サーバ91とソフト開発サーバ92は、上述のサーバ23に対応している。

【0081】このようにリンク情報をたどることで、クライアント21が別のサーバ23にアクセスして、リンク先のファイルのデータを表示する動作は、従来のWWWシステムでも行われていた。

【0082】しかし、本発明では、ただ単にリンク先のオブジェクトのデータを表示するだけでなく、そのメソッドをサーバ23に実行させることができ、さらに、サーバ23からクライアント21へメソッドを通知することもできる。このように、サーバ23間で自動的にメソッド実行を依頼することで、一連の処理がネットワーク透過的に行われる。異なるサーバ23へのメソッド実行の依頼は、オブジェクト間のメッセージ送信を用いたメソッド・コールにより実現される。

【0083】分散システム環境でのオブジェクト間のメッセージ交換の標準仕様としては、OMG (object management group) のCORBA (common object request broker architecture) が知られている。このアーキテクチャでは、クライアントは、データベースや処理プログラムがどこにあるかを意識せずに、ORB (object request broker) に要求を渡せば、ORBが適切な処理をして結果をクライアントに返してくれる。

【0084】ここでは、クライアント21のリンク情報管理モジュール42に保持されたリンク情報を利用することで、異なるサーバ23間にまたがったメソッド・コールが可能となる。

【0085】あるサーバ23に接続しているユーザが新しいオブジェクトを作成する場合は、作成したいオブジェクトのクラスとオブジェクトを保存するサーバ23とを指定して、そのサーバ23のオブジェクト生成モジュール71に処理を依頼する。

【0086】このとき、ユーザは、新しいオブジェクトに対するリンクの情報も同時に指定することができる。あるいはまた、オブジェクト生成71モジュールは、作成されるオブジェクトのクラス情報に基づき、必要なリンク先オブジェクトの指定をユーザに対して要求するようにしてもよい。

【0087】こうして、ユーザによるリンクの指定が終わると、クライアント21は、サーバ23に対してリンク情報を含むオブジェクト情報を送るとともに、管理サーバ61に対して、新しいオブジェクトの登録を通知する。また、そのリンク先のオブジェクトに対しては、リンク情報の追加を要求する。

【0088】また、既に存在するオブジェクトに対して新たに別のオブジェクトとのリンクを設定することもできる。例えば、組織オブジェクトと文書オブジェクトの間に図12に示したような関連を付けることもできるし、新たな文書を作成して、組織オブジェクトに対するリンクを張ることもできる。オブジェクトの削除も同様の手順によって実現できる。図14は、図13の例における新規オブジェクトの作成動作を示している。図14において、例えば、クライアント21がソフト研サーバ91にアクセスし、“ソフト研”の組織オブジェクト93の情報を表示しているとする。図15は、このときのユーザビューを示している。画面上で、下線が引かれた部分は他のオブジェクトへのリンクを表し、そこをクリックすることによって、該当するオブジェクトの情報が表示される。

【0089】この画面上で、ユーザが新規所属員の部分をクリックすると、クライアント21を介して、ソフト開発サーバ92に、新規所属員のオブジェクトの生成が依頼される。このとき、クライアント21の接続管理モジュール41は、まず、ソフト開発サーバ92の接続管理モジュール41から、オブジェクト生成モジュール7

1へのリンク情報“ソフト開発サーバ／／オブジェクト生成”を取得し、それをリンク情報管理モジュール42に格納する。

【0090】新規所属員の定義画面95が表示されると、ユーザは、****で示される部分を必要に応じて定義し、OKボタン96をクリックする。すると、クライアント21は、リンク情報“ソフト開発サーバ／／オブジェクト生成”を用いて、接続中のソフト開発サーバ92のオブジェクト生成モジュール71に対して、新規所属員の個人オブジェクトの作成を要求する。

【0091】これを受けて、オブジェクト生成モジュール71は、所属員“山田”の個人オブジェクト94を生成する。このとき、“山田”の所属組織は“ソフト研”と定義されているので、“ソフト研”の組織オブジェクト93へのリンクが自動的に設定される。

【0092】また、クライアント21は、起動時に管理サーバ61のディレクトリ情報管理モジュール63へのリンク情報“管理サーバ／／ディレクトリ情報管理”を取得して、リンク情報管理モジュール42に保持しており、それを用いて管理サーバ61に新たなディレクトリ情報の追加を依頼する。これにより、管理サーバ61（データベース62）のディレクトリ情報66には、オブジェクト94が新たなオブジェクトとして登録される。

【0093】また、個人オブジェクト94から、ソフト研サーバ91のオブジェクト93へのリンク情報“ソフト研サーバ／／ソフト研”を取得して、オブジェクト93にリンクの追加を依頼する。これにより、オブジェクト93の所属員の項目には、“山田”の個人オブジェクト94へのリンクが張られ、クライアント21の“ソフト研”のオブジェクト93の画面上にも、新しい所属員名“山田”が追加表示される。

【0094】ここでは、新規オブジェクトの作成と、それに伴うディレクトリ情報およびリンクの追加を、クライアント21主導の処理として実現している。しかし、ディレクトリ情報およびリンクの追加は、クライアント21を介して、ソフト開発サーバ92から管理サーバ61およびソフト研サーバ91へ依頼する形で実行することも可能である。

【0095】また、ユーザが新しいサーバ23を登録する際には、管理サーバ61の新サーバ設定モジュール65に処理を依頼する。依頼された新サーバ設定モジュール65は、リンク設定モジュール70、オブジェクト生成モジュール71等の必要なモジュールを新サーバ23にロードする。

【0096】サーバ追加時には、新サーバ設定モジュール65により、図12に示した特殊なオブジェクトクラスであるサーバクラス82のオブジェクトも作成される。ネットワーク25上におけるそのサーバ23のアドレスをこのオブジェクトに格納することで、サーバ23

が管理サーバ61に登録される。アドレスとしては、IP (internet protocol) アドレスやURL等の任意のものが用いられる。

【0097】新サーバ23が登録された時点では、そのサーバ23上にはオブジェクトは存在しないが、管理サーバ61を通して、分散環境上のすべてのオブジェクトにアクセスすることができる。

【0098】次に、図9の各種サービスモジュール73により提供されるユーザ管理、検索等のサービスについて説明する。このサービスモジュール73を設けることで、上述した個人オブジェクトをユーザ管理に利用することができ

る。【0099】新規ユーザが登録される際に、ユーザ管理用の個人オブジェクトを作成し、そこでパスワード等のユーザ情報を保持する。この個人オブジェクトは管理サーバ61で管理され、ユーザが個人オブジェクトの存在しない別のサーバ23からシステムにログインしたとき、管理サーバ61がそのオブジェクトにアクセスして、ユーザ認証等を行う。

【0100】また、オブジェクトとリンクの情報を、ユーザがシステムにログインした際のログイン画面に利用することができる。例えば、個人オブジェクトからは、図12に示したように、組織/プロジェクト、文書、プロセスへのリンクが定義されている。そこで、ログイン画面で、そのユーザの属する組織やプロジェクト、ユーザの文書、割り当てられているプロセスへのリンクを表示することができる。

【0101】図16は、このようなログイン画面の例を示している。この画面上で、ユーザが“ソフト研”をクリックすると、“ソフト研”の組織オブジェクト93の情報が表示され、“ブラウザ開発”または“サーバ開発”をクリックすると、対応するプロジェクトオブジェクトの情報が表示される。また、“部会資料”または“開発仕様書”をクリックすると、対応する文書オブジェクトの情報が表示される。

【0102】さらに、サービスモジュール73は、検索サービスも提供する。オブジェクト間のリンクは、図12のようなクラス間の関係として構成されているので、それをたどることで、ある条件を満たすようなオブジェクトを検索することができる。例えば、組織/プロジェクトクラス83と文書クラス84の間のリンクを参照して、“ソフト研”に属している“議事録”をすべて検索して表示することができる。

【0103】従来のWWWシステムの検索ツールは総当たりの検索を行っているため、検索処理にかなりの時間を要している。これに対して、本発明では、クラス間の関係に基づくインテリジェントな検索が可能であり、検索処理が効率よく行われる。

【0104】以上説明した実施形態では、一般に、ネットワーク25上のすべてのオブジェクトを管理サーバ6

1に登録すると、そのデータ量は膨大になる。そこで、オブジェクトの一部だけを管理サーバ61に登録することにより、ディレクトリ情報66の巨大化による処理効率の劣化を防ぐことを考える。例えば、クラス情報に従って、比較的上位に位置するようなオブジェクトのみを管理サーバ61に登録しておき、それより下位のオブジェクトにはリンクをたどることで到達するようにする。

【0105】図12の例では、例えば、サーバクラス82、組織/プロジェクトクラス83、および個人クラス85のオブジェクトのみを、管理サーバ61に登録する。文書クラス84のオブジェクトは、一般に非常に多くなるが、これには組織/プロジェクトクラス83や個人クラス85のオブジェクトから、リンクをたどってアクセスすることができる。

【0106】さらに、管理サーバ61によるディレクトリ管理自体をなくしてしまうことも可能である。この場合は、図9のレプリケーションモジュール72が、サーバクラス82のオブジェクトの情報のみを、すべてのサーバ23間でレプリケーションする。

【0107】レプリケートされたクラスやオブジェクトの情報は、レプリケーション情報69として各サーバ23のデータベース24に格納される。これにより、ネットワーク25上における各サーバ23の所在場所の情報が、すべてのサーバ23によって共有される。

【0108】また、各オブジェクトには、それが存在するサーバ23のサーバオブジェクトとの間のリンクを持たせておく。これにより、サーバオブジェクトを経由して、各オブジェクトから他のすべてのオブジェクトにアクセスすることが可能となる。

【0109】ネットワーク25上において、サーバ23の追加はデータ更新ほど頻繁には発生しないので、サーバオブジェクトのレプリケーションも、それほど頻繁に行う必要がない。したがって、このレプリケーション処理を直接サーバ23間で行っても、システム全体の効率が大きく低下することはない。

【0110】また、管理サーバ61の負荷が重いときには、ディレクトリ情報66の一部または全部を、各サーバ23へレプリケートすることによって、各オブジェクトへのアクセスを高速化することができる。この処理もまた、レプリケーションモジュール72により行われ、レプリケートされた情報は、レプリケーション情報69としてデータベース24に格納される。

【0111】各オブジェクトについても、同様のレプリケーションが可能であり、それにより、アクセスが高速化される。この場合、原本となるマスターオブジェクトとそのレプリカ(コピー)のオブジェクトの間にはリンクが張られる。

【0112】そして、クライアント21からレプリカの情報の変更が要求されると、自動的にそのマスターオブジェクトを持つサーバ23へ接続が切り替わり、マスタ

一オブジェクトが更新される。その後、レプリケーション機能により、マスターオブジェクトの変更が、そのレプリカである他のオブジェクトに反映される。

【0113】次に、図17および図18を参照しながら、ネットワーク・コンピューティングシステムの動作をより詳細に説明する。図17は、ネットワーク・コンピューティングシステムの一般的な動作を示すフローチャートである。まず、クライアント21のユーザがボタンを押す等のユーザインタラクションが生じると（ステップS11）、必要に応じて、ビューモジュール74がサーバ23上のオブジェクトに対し、メソッドの実行を要求する（ステップS12）。このとき、要求先のオブジェクトの論理ID（サーバ名を含む）は、リンク情報管理モジュール42から検索される。

【0114】次に、クライアント21の接続管理モジュール41は、その論理IDを参照して、要求先のサーバ23に接続しているかどうかを判定し（ステップS13）、そのサーバ23に接続していなければ、接続処理を行ってメソッド実行の要求を通知する（ステップS14）。

【0115】次に、サーバ23の接続管理モジュール41は、オブジェクト管理テーブルを参照して、論理IDをもとに要求先のオブジェクトを検索する（ステップS15）。オブジェクト管理テーブルは、各オブジェクトの論理IDをオブジェクトポインタに変換するための情報を持ち、例えば、接続管理モジュール41内に保持されている。こうして、検索されたオブジェクトが、要求されたメソッドを実行する。

【0116】ここで、メソッド実行中に他のオブジェクトのメソッドの実行要求があるかどうかチェックされ（ステップS16）、そのような要求があれば、接続中のクライアント21に対して、そのメソッドの実行を依頼する（ステップS17）。これにより、ステップS16以降の動作が繰り返される。

【0117】メソッド実行中に他のメソッドの実行要求がなく、メソッド実行が終了すると、サーバ23から関連するクライアント21に対して、各種の通知や情報送信が実行される（ステップS18）。このとき、メソッド実行により変更/追加されたオブジェクトの情報等がクライアント21に送信される。

【0118】クライアント21のビューモジュール74は、受け取った情報を用いてビューを変更し（ステップS19）、必要に応じて、リンク情報管理モジュール42のリンク情報も変更/追加する（ステップS20）。こうして、変更された画面上で、新たなユーザインタラクションが生じると（ステップS11）、ステップS12以降の動作が繰り返される。

【0119】また、図18は、新規オブジェクトの作成処理のフローチャートである。まず、ユーザがクライアント21のビュー上で、図14に示したような新規オブ

ジェクトの作成を指示する（ステップS21）。このとき、クライアント21の接続管理モジュール41は、そのオブジェクトを格納すべきサーバ23のアドレス情報を持っていないければ、接続中のサーバ23または管理サーバ61からそれを取得する（ステップS22）。

【0120】次に、接続管理モジュール41は、オブジェクトを格納すべきサーバ23のオブジェクト生成モジュール71に対して、必要なリンク情報を送り、オブジェクト格納メソッドの実行を要求する（ステップS23）。これを受けて、オブジェクト生成モジュール71は、リンク情報を含む新規オブジェクトを生成して、データベース24に格納し、その論理IDをクライアント21に返す。

【0121】次に、クライアント21は、作成された新規オブジェクトに対してリンクを持つような他のオブジェクトに、新規オブジェクトの論理IDを渡して、リンク追加メソッドの実行を要求する（ステップS24）。この要求を受けたオブジェクトは、その論理IDを新たなリンクとして追加する。

【0122】また、必要に応じて、クライアント21は、管理サーバ61のディレクトリ情報管理モジュール63に対して、新規オブジェクトの登録メソッドの実行を要求する（ステップS25）。これを受けて、ディレクトリ情報管理モジュール63は、新規オブジェクトのクラス名、オブジェクト名、アドレス情報等をディレクトリ情報66に追加し、一連の処理が終了する。

【0123】このように、本発明においては、必要に応じてサーバ側からクライアント側にリンク情報を送ることにより、クライアントを介したメソッド・コールが行われる。したがって、サーバ間で必要となる通信についても、クライアント主導で行うことができ、従来のような複雑な通信制御が不要となる。

【0124】実際、ネットワーク上のサーバのデータ更新は、大抵の場合、クライアントのアクセスにより発生することが多い。このとき、サーバは既にクライアントと接続しているわけであるから、新たに他のサーバと接続を確立するよりも、接続中のクライアントを介して通信の方が合理的であるといえる。本発明は、このようなクライアント主導のサーバ間通信の枠組みを与えている。

【0125】

【発明の効果】本発明によれば、分散システム上での情報の追加、更新、および情報間のリンクの設定等が容易に行えるようになる。したがって、これまで難しかった大きな分散情報システムや企業モデルの構築が可能になる。

【図面の簡単な説明】

【図1】本発明のネットワーク・コンピューティングシステムの原理図である。

【図2】ネットワーク・コンピューティングシステムの

構成図である。

【図3】情報処理装置の構成図である。

【図4】サーバへの接続時の状態を示す図である。

【図5】他のサーバのオブジェクトへのアクセスを示す図である。

【図6】他のサーバへの接続確立後の状態を示す図である。

【図7】第1のサーバ間の情報交換を示す図である。

【図8】第2のサーバ間の情報交換を示す図である。

【図9】モジュール構成を示す図である。

【図10】ディレクトリ情報の例を示す図である。

【図11】クラス定義情報の例を示す図である。

【図12】企業モデルのクラス定義の例を示す図である。

【図13】クライアントによるオブジェクト情報の参照動作を示す図である。

【図14】オブジェクト情報の作成動作を示す図である。

【図15】クライアント画面の例を示す図である。

【図16】ログイン画面の例を示す図である。

【図17】ネットワーク・コンピューティングの動作フローチャートである。

【図18】オブジェクトの新規作成処理のフローチャートである。

【図19】従来のクライアント-サーバシステムの構成図である。

【図20】従来の分散クライアント-サーバシステムの構成図である。

【図21】WWWシステムの動作フローチャートである。

【符号の説明】

1、21 クライアント

2、3、4 専用通信モジュール

5、23、91、92 アプリケーションサーバ

* 6、24、62 データベース

11、13 共通通信手段

12、14 接続管理手段

22 共通通信モジュール

25 ネットワーク

31 CPU

32 メモリ

33 入力装置

34 出力装置

10 35 外部記憶装置

36 媒体駆動装置

37 ネットワーク接続装置

38 バス

39 可搬記録媒体

41 接続管理モジュール

42 リンク情報管理モジュール

43、50、95 ユーザビュー

44、45、93、94 オブジェクト

46、47、48、49、51、52、53、54、9

20 6 ボタン

61 管理サーバ

63 ディレクトリ情報管理モジュール

64 クラス情報管理モジュール

65 新サーバ設定モジュール

66 ディレクトリ情報

67 クラス定義情報

68 オブジェクト情報

69 レプリケーション情報

70 リンク設定モジュール

30 71 オブジェクト生成モジュール

72 レプリケーションモジュール

73 各種サービスモジュール

74 ビューモジュール

* 81、82、83、84、85、86、87 クラス

【図15】

クライアント画面の例を示す図

組織名	ソフト研
組織コード	1234
所属員	山本
	鈴木
新規所属員	

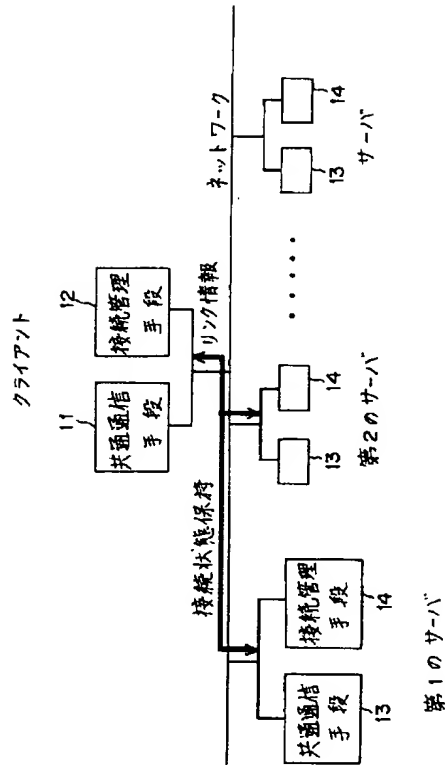
【図16】

ログイン画面の例を示す図

名前	山田太郎
従業員番号	9876
所属組織	ソフト研
所属プロジェクト	ブラウザ開発
	サーバ開発
関連文書	開発資料
	開発仕様書

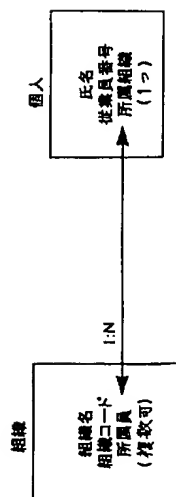
【図1】

本発明の原理図



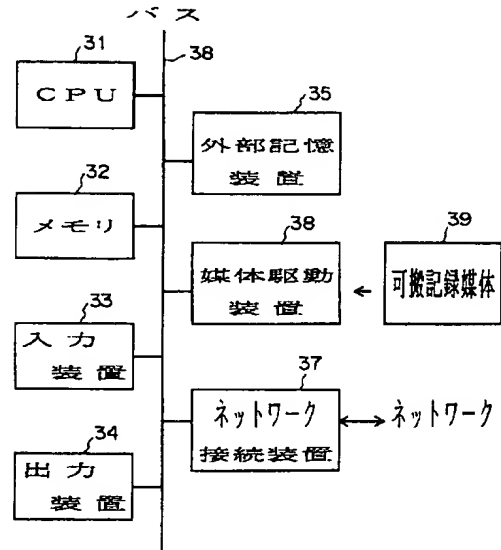
【図11】

クラス定義情報の例を示す図



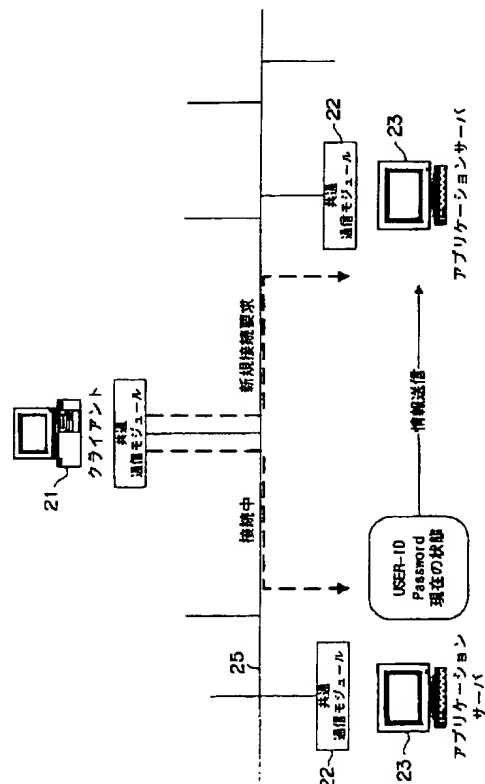
【図3】

情報処理装置の構成図



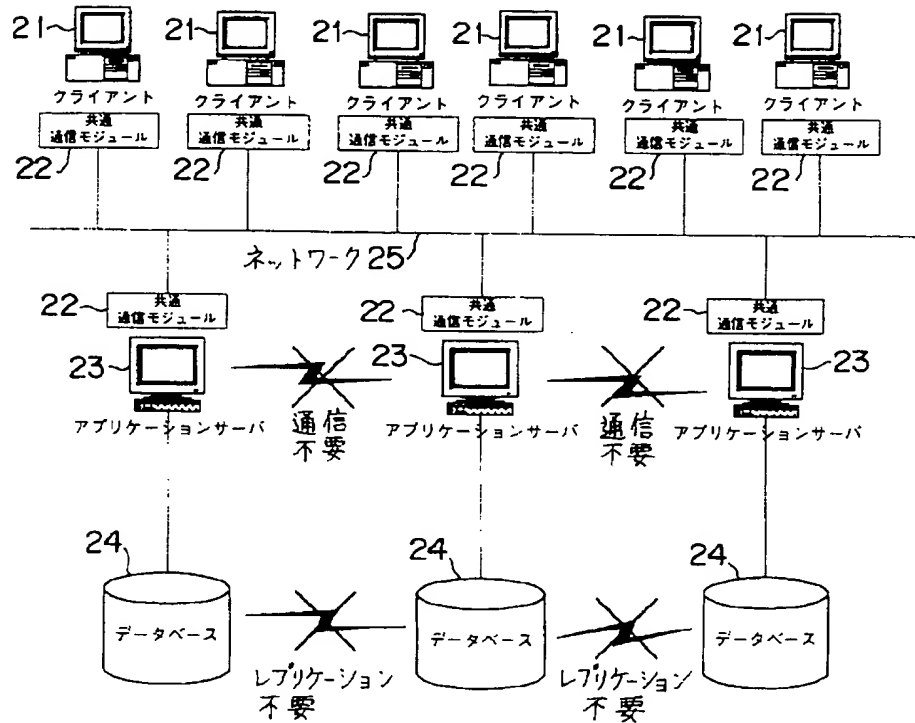
【図7】

第1のサーバ間の情報交換を示す図



【図2】

ネットワーク・コンピューティングシステムの構成図

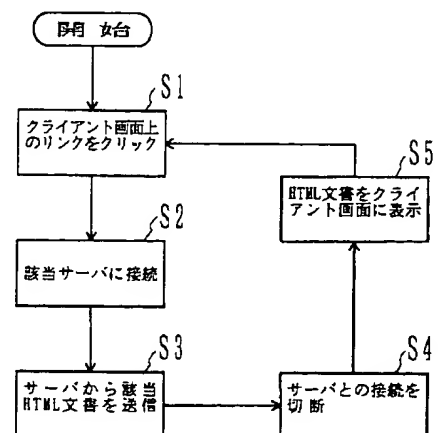


【図10】

ディレクトリ情報の例を示す図

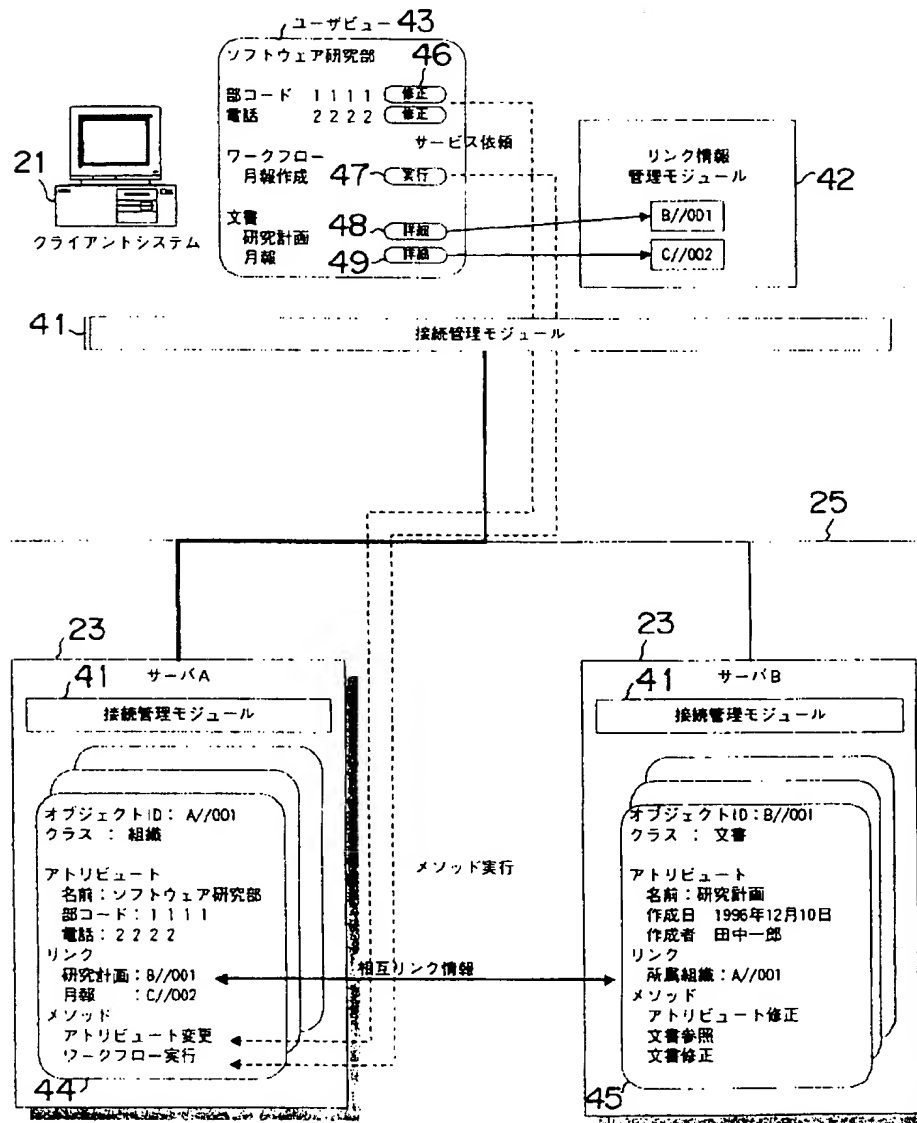
クラス名	オブジェクト名	アドレス
組織	研究所	研究所サーバ// 1001
組織	ソフト研	ソフト研サーバ// 1002
組織	半導体研	研究所サーバ// 1003
個人	山田 太郎	ソフト開発サーバ// 1004
個人	鈴木 健一	研究所サーバ// 1005
サーバ	研究所サーバ	研究所サーバ// 1006
サーバ	ソフト研サーバ	ソフト研サーバ// 1007
サーバ	ソフト開発サーバ	ソフト開発サーバ// 1008
サーバ	半導体研サーバ	半導体研サーバ// 1009

【図21】

WWW システム
の動作フローチャート

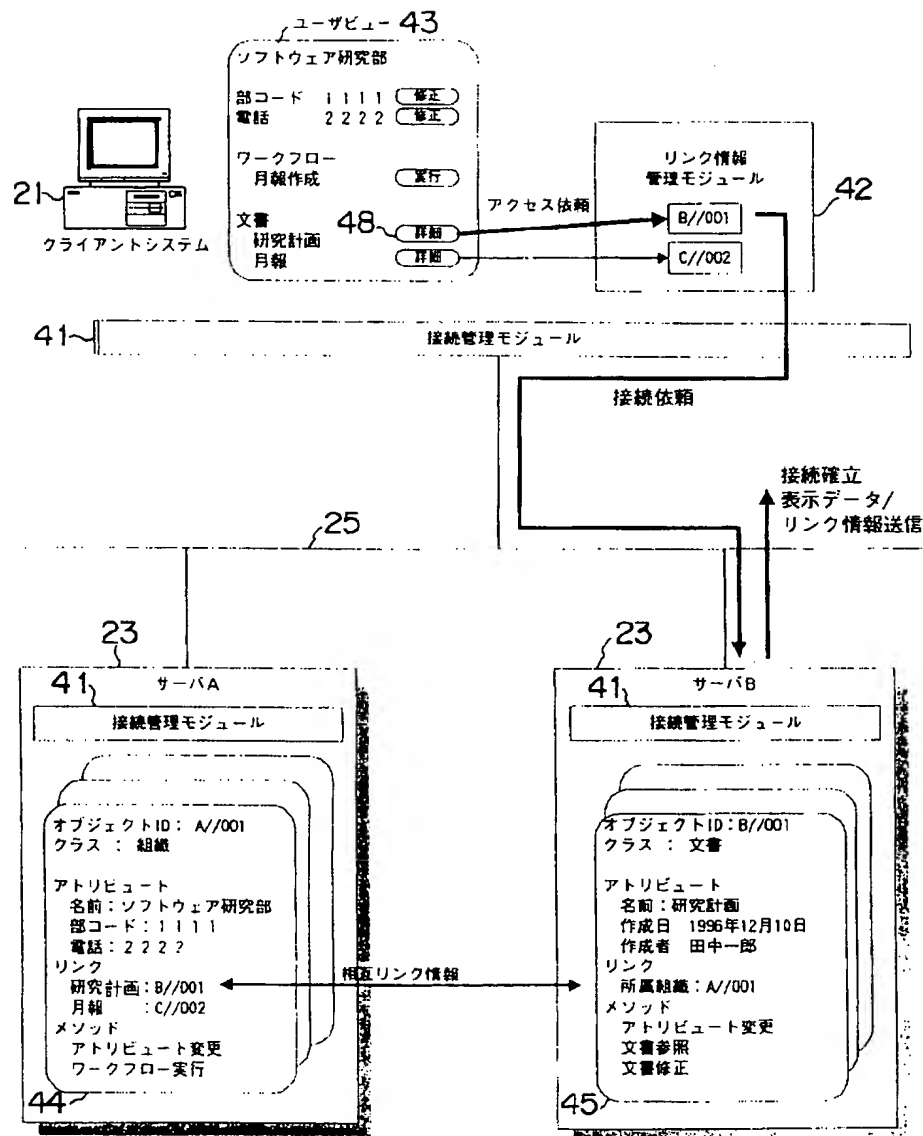
【図4】

サーバへの接続時の状態を示す図



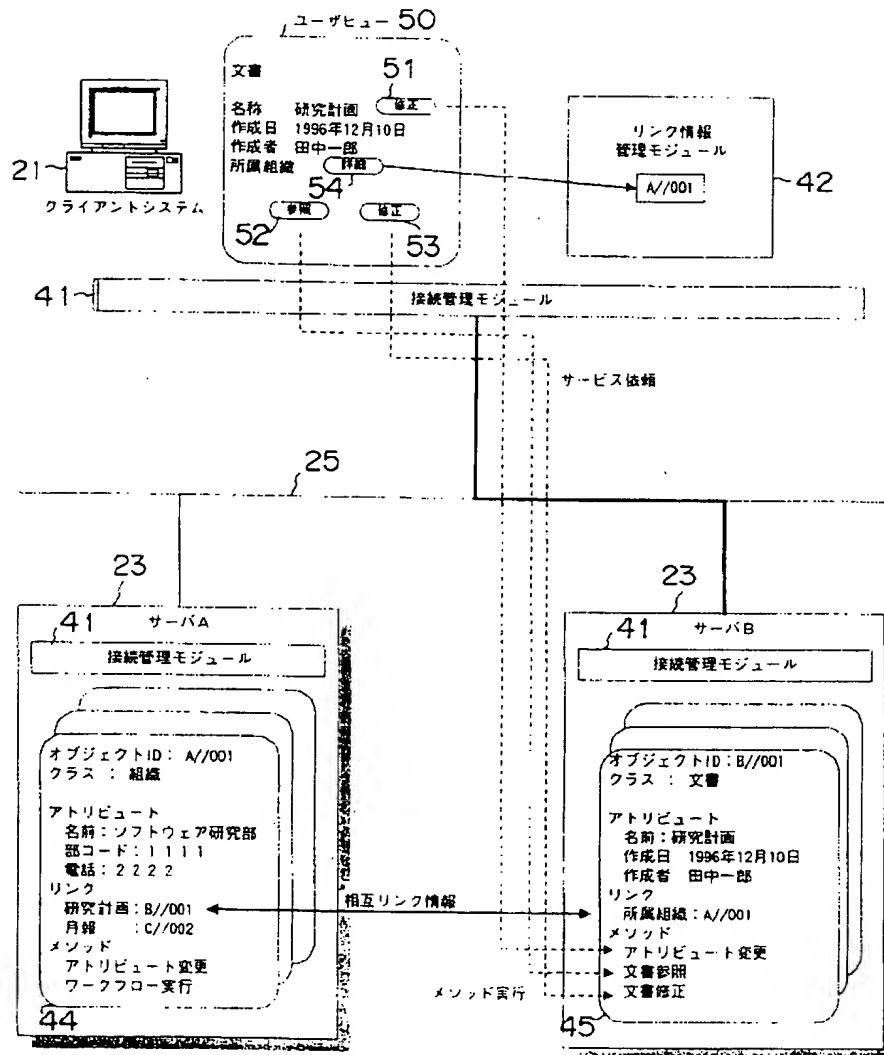
【図5】

他のサーバのオブジェクトへのアクセスを示す図



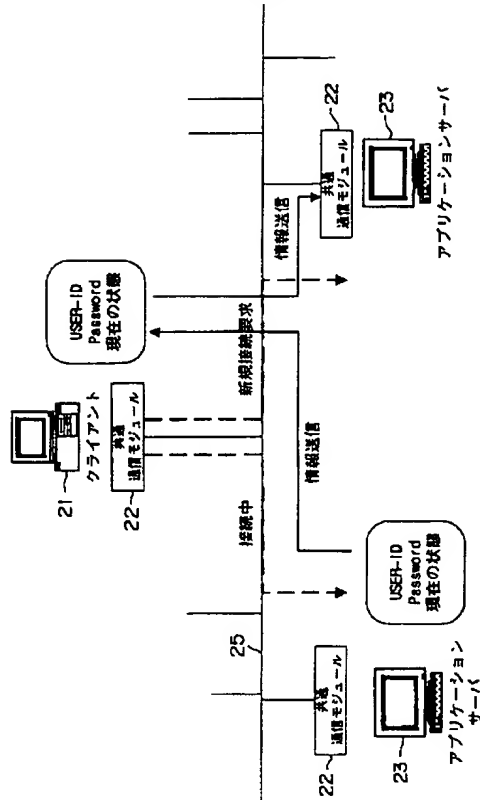
【図6】

他のサーバへの接続確立後の状態を示す図



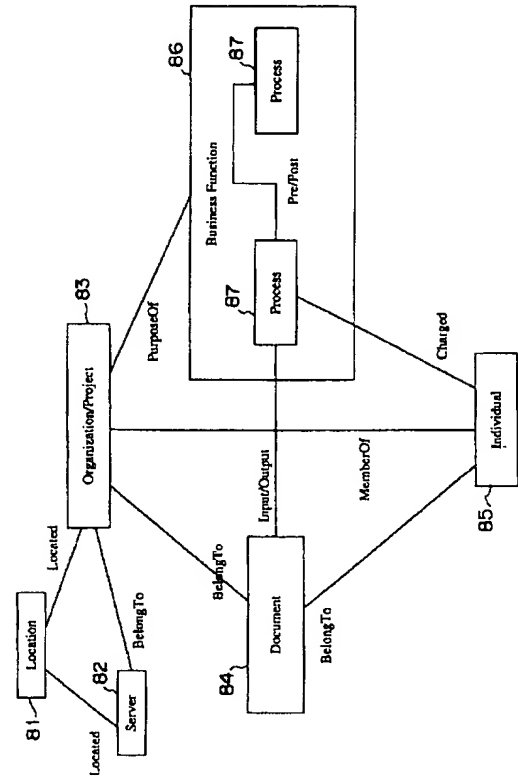
【図8】

第2のサーバ間の情報交換を示す図



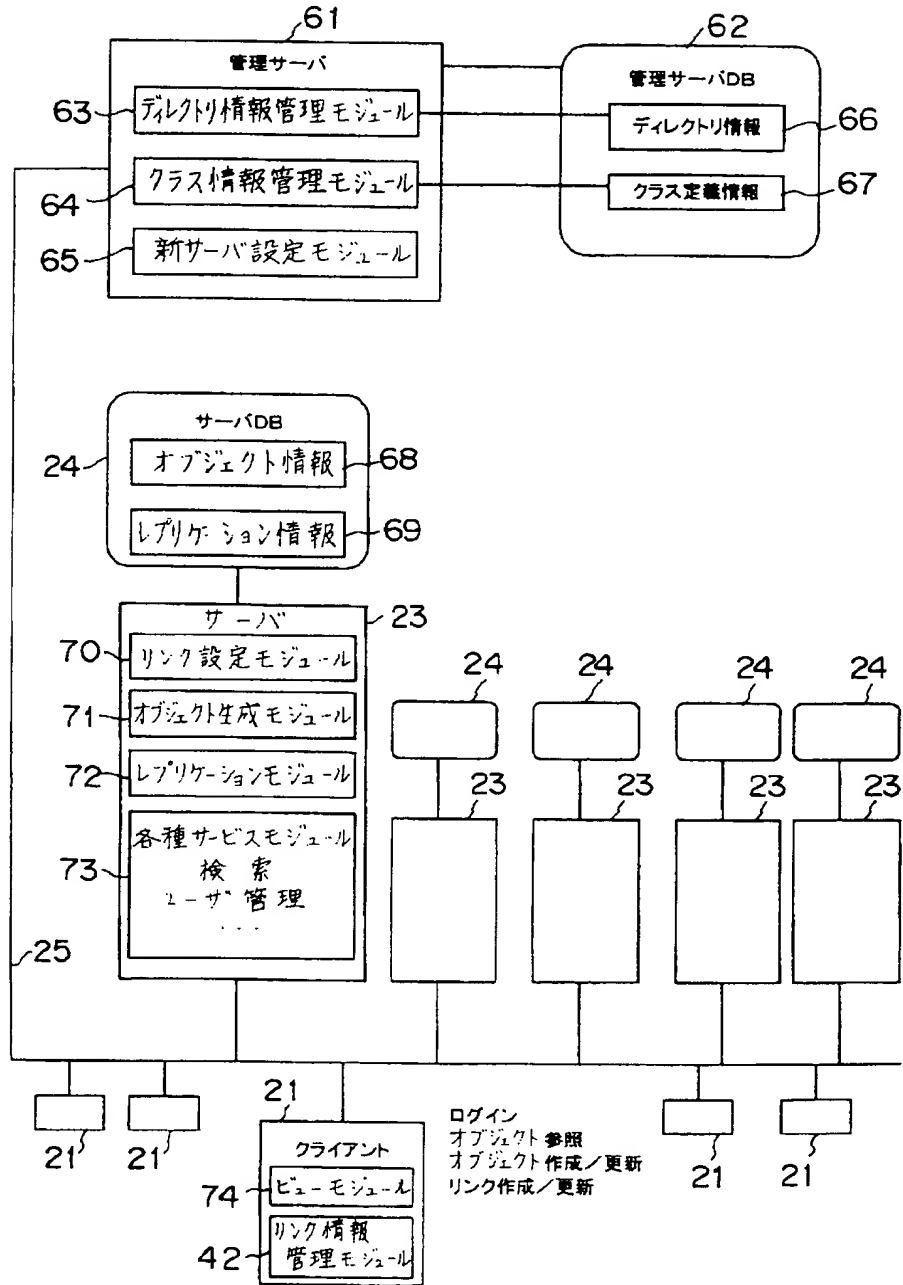
【図12】

企業モデルのクラス定義の例を示す図



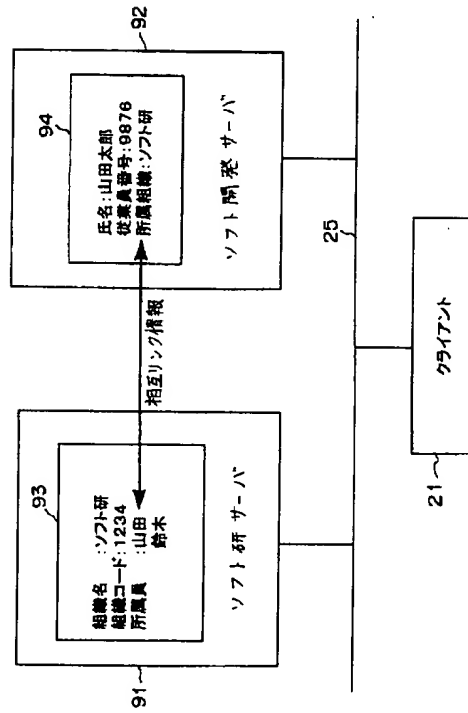
【図9】

モジュール構成を示す図



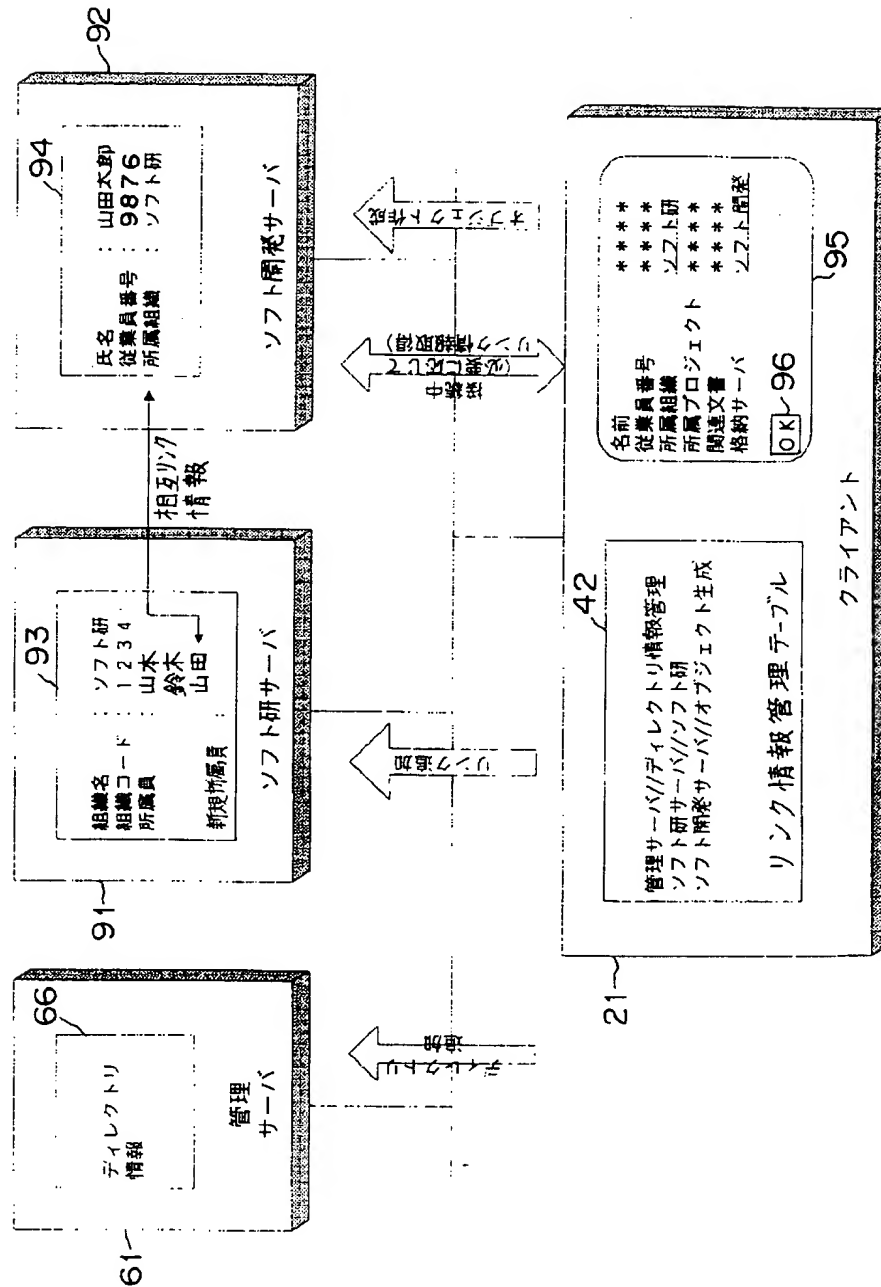
【図13】

クライアントによるオブジェクト情報の
参照動作を示す図



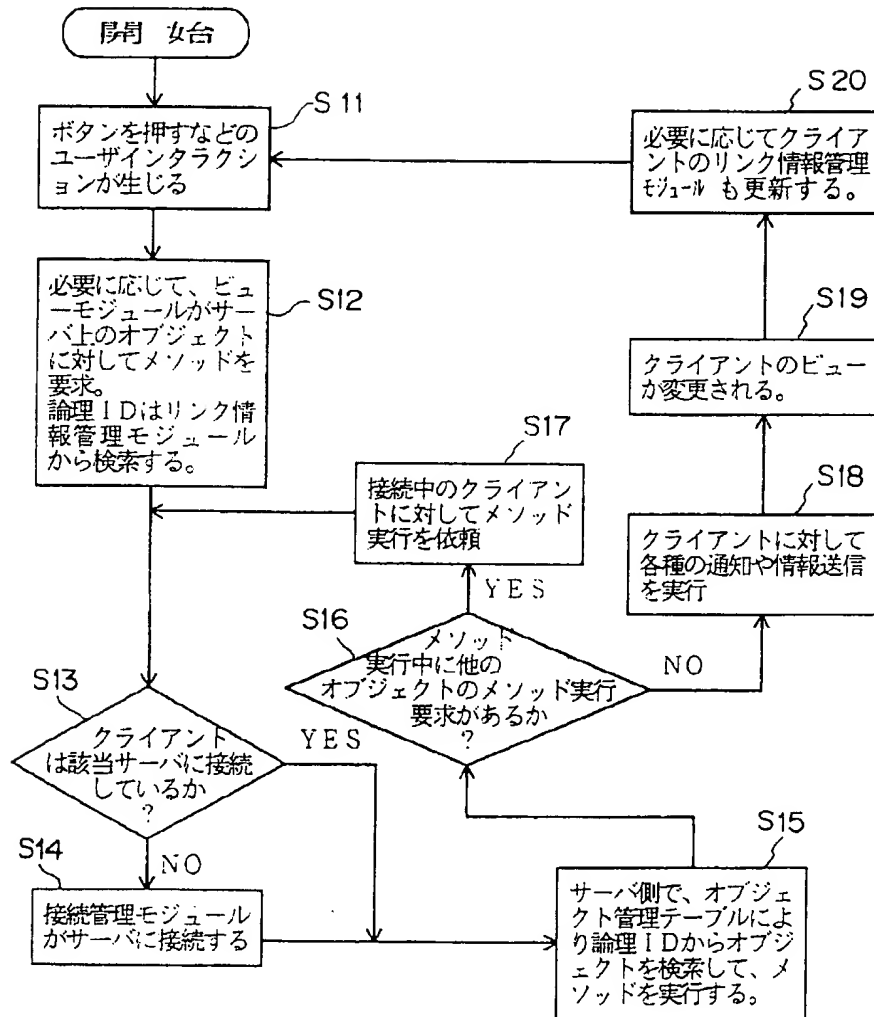
【図14】

オブジェクト情報の作成動作を示す図



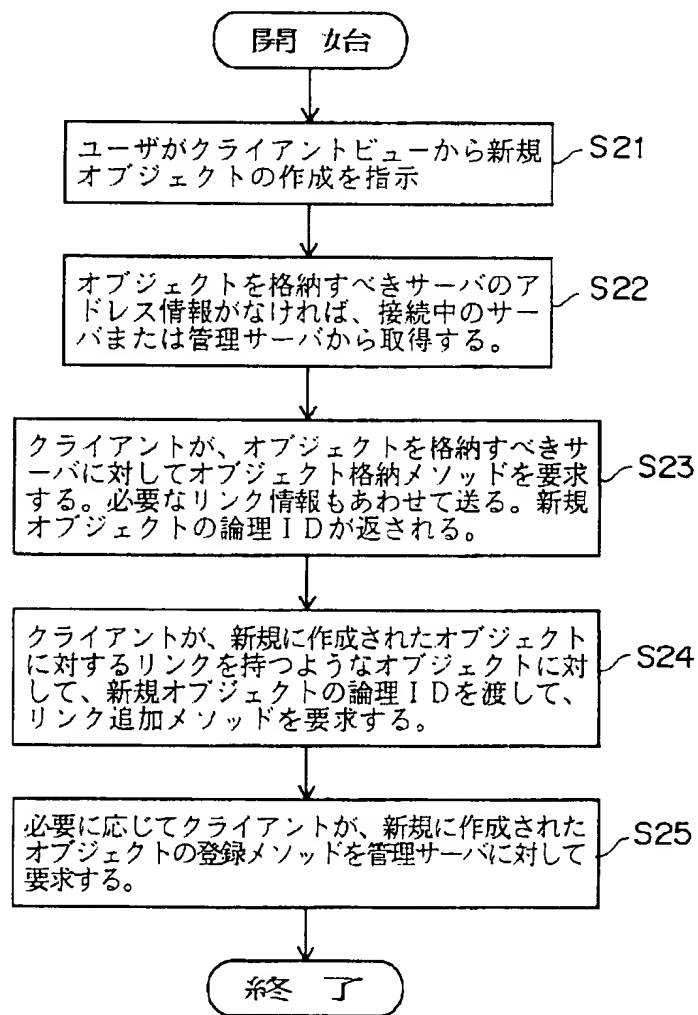
【図17】

ネットワーク・コンピューティングの動作フローチャート



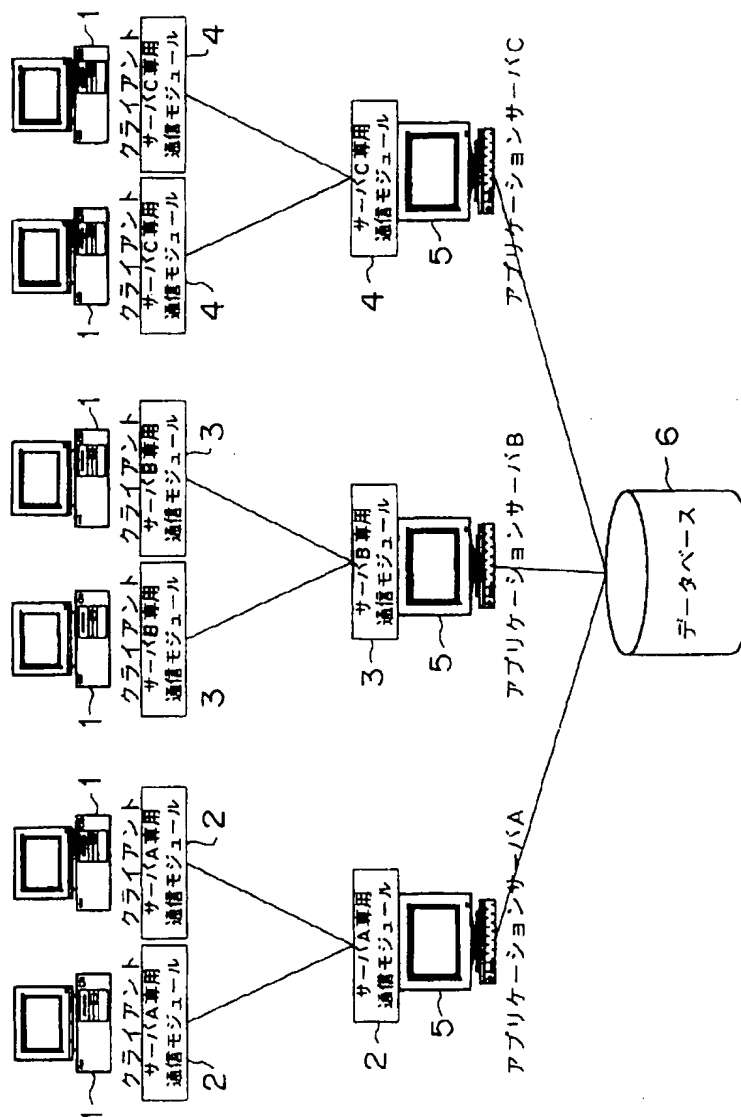
【図18】

オブジェクトの新規作成処理のフローチャート



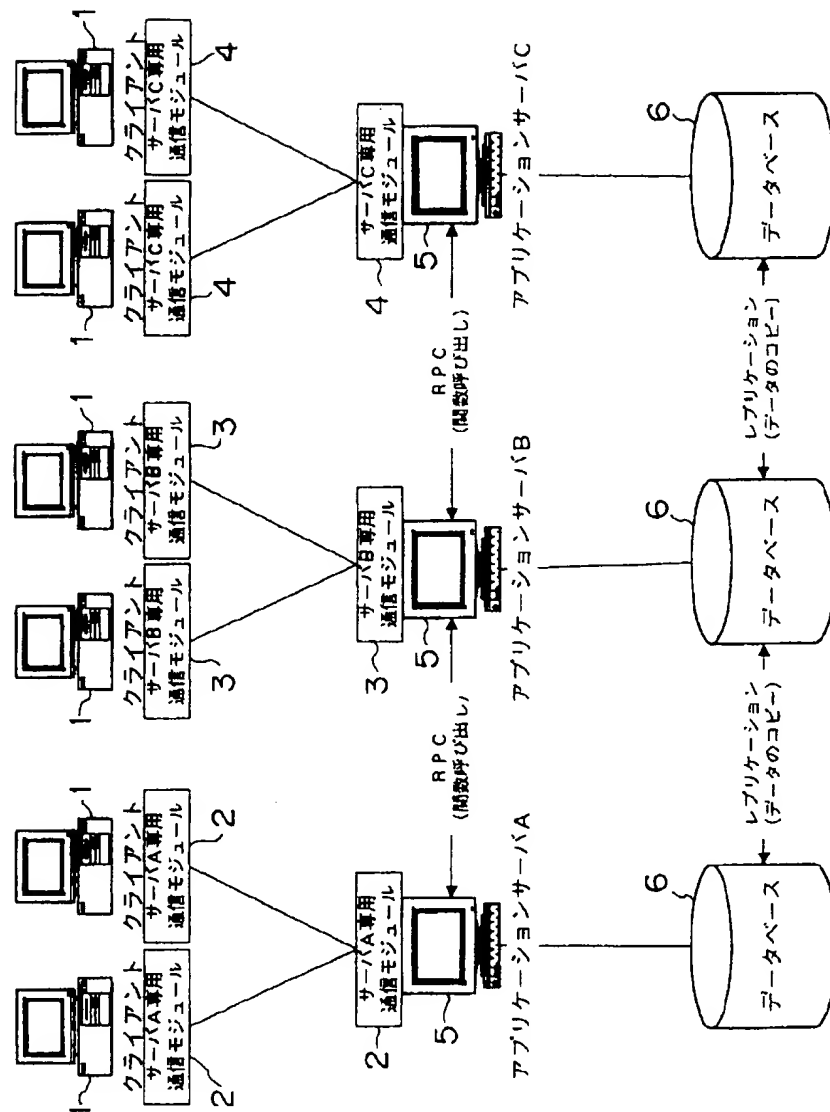
【図19】

従来のクライアント-サーバシステムの構成図



【図20】

従来の分散クライアント-サーバシステムの構成図



フロントページの続き

(72)発明者 金谷 延幸
 神奈川県川崎市中原区上小田中4丁目1番
 1号 富士通株式会社内

(72)発明者 大久保 隆夫
 神奈川県川崎市中原区上小田中4丁目1番
 1号 富士通株式会社内

(72)発明者 上原 三八

神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.